

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)	Group Art Unit: 2176
)	
Falko Tesch et al.)	Examiner: Paul H. Nguyen Ba
)	
Application No. 09/929,742)	
)	
Filed: August 13, 2001)	
)	
For: METHODS AND SYSTEMS FOR)	
PROCESSING EMBEDDED OBJECTS)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT “D” AFTER FINAL

Dear Sir:

This Amendment “D” After Final is filed in response to the final Office Action mailed on April 17, 2006. Applicants respectfully request amendment of the patent application, and reconsideration and allowance of the pending claims.

IN THE CLAIMS

This listing of claims replaces all prior listings.

1. (Currently Amended) A method in a data processing system for processing a document containing an embedded object having a first format corresponding to a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and
while the document is being loaded into the memory,

identifying the embedded object contained in the document,

automatically determining whether the first program is an unavailable program;

and

when it is determined that the first program is an unavailable program, automatically converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system.

2. (Original) The method of claim 1, further comprising the step of:
receiving an indication of the second format from a user.

3. (Original) The method of claim 2, wherein the data processing system contains a plurality of programs, each with an associated format, and wherein the method further comprises the steps of:

determining which of the plurality of programs are available on the data processing system; and

displaying the associated formats of the available programs to the user.

4. (Original) The method of claim 1, further comprising the steps of:
receiving an indication of a third format from a user;
converting the embedded object into the third format; and
storing the embedded object.

5. (Original) The method of claim 1, wherein the converting step includes automatically identifying the second format.

6. (Original) The method of claim 1, further comprising the step of:
converting the embedded object into an intermediate format prior to converting the embedded object into the second format.

7. (Currently Amended) A method in a data processing system for processing a document containing an embedded object having a first format corresponding to a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and
while the document is being loaded into the memory,
identifying the embedded object contained in the document,
automatically determining whether the first program is an unavailable program;
and
when it is determined that the first program is an unavailable program,
automatically converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system;
receiving an indication of a third format from a user;
converting the embedded object into the third format; and
storing the embedded object in the third format.

8. (Currently Amended) A method in a data processing system containing a plurality of programs, each with an associated format, the data processing system for processing a document containing an embedded object having an originating format corresponding to an originating program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and
while the document is being loaded into the memory,
identifying the embedded object contained in the document,
automatically determining whether the originating program is unavailable;

when it is determined that the originating program is unavailable,
determining which of the plurality of programs are available on the data processing system,
displaying the associated formats of the available programs to a user, and
receiving an indication of a selected one of the displayed formats from the user; and
automatically converting the embedded object into the selected format.

9. (Previously Presented) A method in a data processing system for processing a document containing an embedded object having a first format corresponding to a first program, the data processing system having a memory, the method comprising the steps of:
initiating loading of the document into the memory;
while the document is being loaded,
determining whether the first program is unavailable,
when it is determined that the first program is unavailable,
automatically converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system;
after the document is loaded, receiving a request from the user to edit the embedded object; and
responsive to receiving the request from the user, converting the embedded object into a third format that is suitable for use with a third program that is available on the data processing system.

10. (Currently Amended) A data processing system comprising:
a secondary storage device comprising a target document containing an embedded object having a first format corresponding to a first program;
a memory comprising a computer program that initiates loading of a document into a memory of the data processing system, and, while the document is being loaded, identifies the embedded object in the document, automatically determines whether the first program is an unavailable program, and, when it is determined that the first program is an unavailable program,

automatically converts the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system; and
a processing unit that runs the computer program.

11. (Currently Amended) A computer-readable medium containing instructions that cause a data processing system to perform a method for processing a document containing an embedded object having a first format corresponding to a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and
while the document is being loaded into the memory,

identifying the embedded object contained in the document,

automatically determining whether the first program is an unavailable program;
and

when it is determined that the first program is an unavailable program,
automatically converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system.

12. (Original) The computer-readable medium of claim 11, further comprising the step of:

receiving an indication of the second format from a user.

13. (Original) The computer-readable medium of claim 12, wherein the data processing system contains a plurality of programs, each with an associated format, and wherein the method further comprises the steps of:

determining which of the plurality of programs are available on the data processing system; and

displaying the associated formats of the available programs to the user.

14. (Original) The computer-readable medium of claim 11, further comprising the steps of:

receiving an indication of a third format from a user;
converting the embedded object into the third format; and
storing the embedded object.

15. (Original) The computer-readable medium of claim 11, wherein the converting step includes automatically identifying the second format.

16. (Original) The computer-readable medium of claim 11, further comprising the step of:

converting the embedded object into an intermediate format prior to converting the embedded object into the second format.

17. (Currently Amended) A computer-readable medium containing instructions that cause a data processing system to perform a method for processing a document containing an embedded object having a first format corresponding to a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system;

while the document is being loaded into the memory,

identifying the embedded object contained in the document,

automatically determining whether the first program is an unavailable program;

and

when it is determined that the first program is an unavailable program, automatically converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system;

receiving an indication of a third format from a user;
converting the embedded object into the third format; and
storing the embedded object in the third format.

18. (Currently Amended) A computer-readable medium containing instructions that cause a data processing system containing a plurality of programs, each with an associated

format, to perform a method for processing a document containing an embedded object having an originating format corresponding to an originating program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and
while the document is being loaded into the memory,
identifying the embedded object contained in the document,
automatically determining whether the originating program is unavailable; and
when it is determined that the originating program is unavailable,
automatically determining which of the plurality of programs are available
on the data processing system,
displaying the associated formats of the available programs to a user, ~~and~~
receiving an indication of a selected one of the displayed formats from the
user; ~~and~~
converting the embedded object into the selected format.

19. (Currently Amended) A computer-readable medium containing instructions that cause a data processing system to perform a method for processing a document containing an embedded object having a first format corresponding to a first program, the data processing system having a memory, the method comprising the steps of:

initiating loading of the document into the memory;
while the document is being loaded,
determining whether the first program is unavailable, and
when it is determined that the first program is unavailable, automatically
converting the embedded object into a second format different from the first format that is
suitable for use with a second program that is available on the data processing system;
after the document is loaded, receiving a request from the user to edit the embedded
object; and
responsive to receiving the request from the user, converting the embedded object into a
third format that is suitable for use with a third program that is available on the data processing
system.

20. (Currently Amended) A computer-readable memory device encoded with a data structure with entries, each entry reflecting embedded data in a document that is accessed by a host program which is encoded on the memory device and which is run by a processor in a system, each entry comprising:

a storage area in which is stored a first identifier of an original program that was utilized during creation of the embedded data and in which is stored a second identifier of an available program to be used for automatically converting the embedded data and accessing the embedded data when the original program becomes unavailable in the system,

wherein the available program is different than the original program, and
wherein, while the document is being loaded into the memory, the host program:

identifies the embedded object contained in the document,
determines whether the first program is unavailable, and
when it is determined that the first program is unavailable, automatically converts
the embedded object into a second format different from the first format that is suitable for use
with a second program that is available on the data processing system.

21. (Original) The computer-readable memory device of claim 20, wherein the second identifier replaces the first identifier.

22. (Original) The computer-readable memory device of claim 20, wherein the original program becomes unavailable because the original program is not installed on the system.

23. (Original) The computer-readable memory device of claim 20, wherein each entry includes a second storage area in which is stored the embedded data in a format suitable for use with the original program and in which is stored the embedded data in a format suitable for use with the available program.

24. (Original) The computer-readable memory device of claim 23, wherein the embedded data in the format suitable for use with the available program replaces the embedded data in the format suitable for use with the original program.

25. (Currently Amended) A data processing system for processing a document containing an embedded object having a first format corresponding to a first program, the data processing system comprising:

means for initiating loading of the document into a memory of the data processing system; and

means for, while the document is being loaded into the memory,

identifying the embedded object contained in the document,

automatically determining whether the first program is an unavailable program;

and

~~means for~~ automatically converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system, when it is determined that the first program is an unavailable program.

26. (Currently Amended) A method in a data processing system comprising a document with data in a native format and with embedded data in a nonnative format, the embedded data suitable for use with a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and

while the document is being loaded into the memory,

identifying the embedded data contained in the document,

receiving an indication of a different format suitable for use with a different program; and

automatically converting the embedded data into the different format, when it is determined that the first program is an unavailable program;

~~wherein the steps of the method are performed while the document is being loaded into memory.~~

27. (Original) The method of claim 26, wherein the receiving step includes the step of:

receiving the indication from a user.

28. (Original) The method of claim 26, wherein the receiving step includes the step of:

retrieving the indication from storage.

29. (Cancelled)

30. (Cancelled)

31. (Currently Amended) A computer-readable medium containing instructions that cause a data processing system to perform a method, the data processing system comprising a document with data in a native format and with embedded data in a nonnative format, the embedded data suitable for use with a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system; and
while the document is being loaded into the memory,

identifying the embedded data contained in the document,

receiving an indication of a different format suitable for use with a different program; and

automatically converting the embedded data into the different format, when it is determined that the first program is an unavailable program;

~~wherein the steps of the method are performed while the document is being loaded into memory.~~

32. (Original) The computer-readable medium of claim 31, wherein the receiving step includes the step of:

receiving the indication from a user.

33. (Original) The computer-readable medium of claim 31, wherein the receiving step includes the step of:

retrieving the indication from storage.

34. (Cancelled)

35. (Cancelled)

36. (Currently Amended) A method in a data processing system for processing a document containing an embedded object having a first format corresponding to a first program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system;

while the document is being loaded into the memory,

identifying the embedded object contained in the document,

determining whether the first program is an unavailable program; and

when it is determined that the first program is an unavailable program, converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system; and

selecting a user selectable setting comprising at least a first setting for automatically performing the step of converting while the document is being loaded into memory and a second setting for performing the step of converting upon selection of the document for editing.

37. (Currently Amended) A computer-readable medium containing instructions that cause a data processing system to perform a method, the data processing system comprising a document with data in a native format and with embedded data in a nonnative format, the embedded data suitable for use with a program, the method comprising the steps of:

initiating loading of the document into a memory of the data processing system;

while the document is being loaded into the memory,

identifying the embedded object contained in the document,

determining whether the first program is an unavailable program; and

when it is determined that the first program is an unavailable program, converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system; and

selecting a user selectable setting comprising at least a first setting for automatically performing the step of converting while the document is being loaded into memory and a second setting for performing the step of converting upon selection of the document for editing.

38. (Currently Amended) A data processing system for processing a document containing an embedded object having a first format corresponding to a first program, the data processing system comprising:

means for initiating loading of the document into a memory of the data processing system;

means for, while the document is being loaded into the memory,

identifying the embedded object contained in the document,

determining whether the first program is an unavailable program; and

~~means for~~ converting the embedded object into a second format different from the first format that is suitable for use with a second program that is available on the data processing system when it is determined that the first program is an unavailable program; and

means for selecting a user selectable setting comprising at least a first setting for automatically performing the step of converting while the document is being loaded into memory and a second setting for performing the step of converting upon selection of the document for editing.

REMARKS

Claims 1-28, 31-33, and 36-38 are pending in the application. In the Final Office Action dated April 17, 2006, the Examiner made the following disposition:

- A.) Rejected claims 1-5, 7-15, 17-19, 25-28, and 31-33 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Oracle Forms® Advanced Techniques, ch. 10, pp. 1-18 ("Oracle")* in view of *Lee, et al. (U.S. Patent No. 6,061,696) ("Lee")*.
- B.) Rejected claims 6, 16, and 20-24 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Oracle* in view of *Francis, et al. (U.S. Patent No. 6,182,092) ("Francis")* and further in view of *Lee*.
- C.) Rejected claims 36-38 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Oracle* in view of *Laverty, et al. (U.S. Patent No. 6,396,593) ("Laverty")* and further in view of *Lee*.

Applicants respectfully traverse the rejections and address the Examiner's disposition below.

- A.) Rejection of claims 1-5, 7-15, 17-19, 25-28, and 31-33 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Oracle Forms® Advanced Techniques, ch. 10, pp. 1-18 ("Oracle")* in view of *Lee, et al. (U.S. Patent No. 6,061,696) ("Lee")*;

Applicants respectfully disagree with the rejection.

Independent claims 1, 7, 8, 10, 11, 17-21, 25, 26, 31, and 36-38 have each been amended to clarify that loading of the document into a memory of the data processing system is initiated, and that, while the document is being loaded into the memory, it is identified whether the embedded object/data is contained in the document and the embedded object/data is automatically converted into a different format when it is determined that the first program is an unavailable program.

Applicants' independent claims 1, 7-11, 17-19, 25, 26, and 31, each as amended, each claim subject matter relating to initiating loading a document into memory. While the document is being loaded into memory, it is identified whether an embedded object/data is contained in the document. And also while the document is being loaded into memory, when it is determined that the first program is an unavailable program, the embedded object/data is automatically converted from a first format, which corresponds to a first program, to a second format that is different from the first format and suitable for use with a second program.

Thus, Applicants' claimed invention inventively converts an embedded object/data within a document "on-the-fly" while the document is being loaded into memory. This provides a seamless and conversion that may not require user interaction.

This is clearly unlike *Oracle* in view of *Lee*, which fails to disclose or suggest automatically converting an embedded data in a document from a first format that is suitable for use with a first program into a second format that is suitable for use with a second program, while the document is being loaded into memory. As acknowledged by the Examiner, *Oracle* fails to disclose or suggest automatically converting embedded data. (*Office Action of 10/18/2005*, page 3). Instead, *Oracle* merely allows a user to manually initiate conversion of an OLE object to a new format, after the related document has been loaded into memory, by selecting the convert option on the Object submenu of the OLE popup menu. (*Oracle*, page 17).

Lee also fails to disclose or suggest automatically converting an embedded data while the related document is being loaded into memory. *Lee* discloses a system that allows a user to edit objects either in their native format or in a standard format. When a user selects an object, *Lee* checks the file types associated with the standard version and the native version of the object. *Lee* then presents editors that may be used to edit the standard version or native version of the object. A user manually selects the desired editor, edits the object, then saves the edits. Edited objects can be automatically converted between standard and native versions. *Lee* 5:55-6:65. Conversion occurs after the document has already been loaded into memory.

The Examiner cites *Oracle* pages 17 and 18 as support for the Examiner's argument that *Oracle* converts an embedded data within a document while the document is being loaded into memory. However, that passage does not make such a teaching. Instead, *Oracle* pages 17 and 18 clearly describe how a user can manually convert an OLE object by inputting different selections into a Convert dialog box. This is clearly performed after a document has been loaded into memory. Nowhere does *Oracle* describe converting an embedded data while a document is being loaded into memory.

Thus, *Oracle* in view of *Lee* fails to disclose or suggest claims 1, 7-11, 17-19, 25, 26, and 31.

Claims 2-5, 12-15, 27, 28, 30, 32, 33, and 35 depend directly or indirectly from claims 1, 11, 26, or 31 and are therefore allowable for at least the same reasons that claims 1, 11, 26, and 31 are allowable.

Applicants respectfully submit the rejection has been overcome and request that it be withdrawn.

B.) Rejection of claims 6, 16, and 20-24 under 35 U.S.C. §103(a) as allegedly being unpatentable over Oracle in view of Francis, et al. (U.S. Patent No. 6,182,092) ("Francis") and further in view of Lee:

Applicants respectfully disagree with the rejection.

Regarding claims 6 and 16:

Applicants' independent claims 1 and 11 are allowable over Oracle in view of Lee as discussed above. Francis still fails to disclose or suggest automatically converting embedded data from a first format, which corresponds to a first program, to a second format while a document is being loaded into memory. Referring to Francis's Figure 8, Francis analyzes a document to determine whether the document contains Hypertext Markup Language (HTML) elements (step 252). If the document contains HTML elements, then the HTML elements are converted to a Rich Text Format (RTF) stream (steps 258, 262, 268, and 270).

Unlike Applicants' claims 1 and 11, nowhere does Francis disclose or suggest automatically converting embedded data from a first format, which corresponds to a first program, to a second format when it is determined that the first program is an unavailable program. Francis makes no such determination. Instead, Francis merely converts elements in a document from one format to another, without determining whether a corresponding program is available, and after the document has been loaded into memory. Thus, Oracle in view of Francis and further in view of Lee still fails to disclose or suggest claims 1 and 11.

Claims 6 and 16 depend directly or indirectly from claims 1 or 11 and are therefore allowable for at least the same reasons that claims 1 and 11 are allowable.

Regarding claims 20-24:

Applicants' independent claim 20, as amended, also claims automatically converting embedded data in a document while the document is being loaded into memory.

As discussed above with reference to claims 1 and 11, *Oracle* in view of *Francis* and further in view of *Lee* fails to disclose or suggest automatically converting embedded data while a document is being loaded into memory. Thus, for at least this reason, *Oracle* in view of *Francis* and further in view of *Lee* fails to disclose or suggest claim 20.

Claims 21-24 depend directly or indirectly from claim 20 and are therefore allowable for at least the same reasons that claim 20 is allowable.

Applicants respectfully submit the rejection has been overcome and request that it be withdrawn.

C.) Rejection of claims 36-38 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Oracle* in view of *Lavery, et al. (U.S. Patent No. 6,396,593) ("Lavery")* and further in view of *Lee*:

Applicants respectfully disagree with the rejection.

Applicants' independent claims 36-38, each as amended, each claim subject matter relating to automatically converting an embedded object, while a document is being loaded into memory, from a first format to a second format when it is determined that a first program corresponding to the first format is an unavailable program.

As discussed above with reference to claims 1 and 11, *Oracle* in view of *Lee* fails to disclose or suggest automatically converting an embedded object while a document is being loaded into memory. Thus, for at least this reason, *Oracle* in view of *Lee* fails to disclose or suggest claims 36-38.

Lavery also fails to disclose or suggest automatically converting an embedded object while a document is being loaded into memory. Thus, *Oracle* in view of *Lavery* and further in view of *Lee* still fails to disclose or suggest claims 36-38.

Applicants respectfully submit the rejection has been overcome and request that it be withdrawn.

CONCLUSION

In view of the foregoing, Applicants submit that the application is in condition for allowance. Notice to that effect is respectfully requested.

Respectfully submitted,

Dated: June 14, 2006

By: /Christopher P. Rauch/
Christopher P. Rauch
Registration No. 45,034
SONNENSCHNEIDER NATH & ROSENTHAL LLP
P.O. Box 061080
Wacker Drive Station, Sears Tower
Chicago, Illinois 60606-1080
(312) 876-8000
Customer No. 58328